



# 一类动态不确定环境下机器人的 滚动路径规划<sup>1)</sup>

席裕庚 张纯刚

(上海交通大学自动化研究所 上海 200030)

(E-mail: Ygx1@mail.sjtu.edu.cn, cgzhang925@online.sh.cn)

**摘要** 研究了一类全局环境未知且存在动态障碍物情况下的机器人路径规划问题. 借鉴预测控制滚动优化原理, 给出了基于滚动窗口的移动机器人路径规划方法. 充分利用机器人实时测得的局部环境信息, 实施在线滚动规划, 把优化与反馈机制合理结合起来, 对动态不确定环境具有良好的适应性. 同时还对滚动规划算法的安全性和可达性进行了分析.

**关键词** 机器人路径规划, 滚动规划, 安全性, 可达性

**中图分类号** TP24

## ROLLING PATH PLANNING OF MOBILE ROBOT IN A KIND OF DYNAMIC UNCERTAIN ENVIRONMENT

XI Yu-Geng ZHANG Chun-Gang

(Institute of Automation, Shanghai Jiaotong University, Shanghai 200030)

(E-mail: Ygx1@mail.sjtu.edu.cn, cgzhang925@online.sh.cn)

**Abstract** In this paper, robot path planning in globally unknown environment with moving obstacles is studied. Using the rolling optimization concept adopted in predictive control, a new strategy of path planning based on rolling windows for a mobile robot is proposed. The method makes full use of the real-time local environmental information detected by the robot and the path planning is performed on-line in a rolling style. Optimization and feedback are combined in a reasonable way. It is well suitable to dynamic uncertain environment. The safety and accessibility of the planning algorithm are also discussed in the paper.

**Key words** Robot path planning, rolling plan, safety, accessibility

1) 国家“973”计划基金(G1998030415)、国家“863”计划基金(9935-02)及国家自然科学基金(69774004, 69934020)资助

收稿日期 2000-04-29 收修改稿日期 2001-05-14

## 1 引言

移动机器人路径规划问题是指在有障碍物的工作环境中,如何寻找一条从给定起点到终点的适当的运动路径,使机器人在运动过程中能安全、无碰撞地绕过所有的障碍物.当机器人具备全局环境信息时,可用一次性的全局规划来得到一条自起点到终点的安全路径,并能对运行过程中的某些性能指标进行优化,这方面的研究早已有广泛的报道<sup>[1~3]</sup>.但实际上,机器人对于环境信息,特别是动态障碍物的信息很难具有先验知识,这时不能离线作出一次性全局规划.机器人只能依靠实时探测到的环境信息,经过多次重规划(滚动规划)来得到可行的安全路径.规划中的安全性和可达性是要考虑的两个主要问题.人们已提出了不少解决未知环境下路径规划问题的方法和策略<sup>[4~8]</sup>,但对于存在动态障碍物的未知环境,还缺少有效的规划方法同时保证安全性和可达性.即使机器人通过漫游对环境信息进行预探测<sup>[9~11]</sup>,至多也只能获取静态的障碍信息,要在动态不确定环境中进行有效规划仍具有很大难度.

本文借鉴预测控制滚动优化原理<sup>[12]</sup>,提出了动态不确定环境下基于滚动窗口的移动机器人路径规划方法.该方法充分利用机器人实时测得的局部环境信息,以滚动方式进行在线规划,实现了优化与反馈的合理结合,具有良好的避碰能力;并且在一定的环境约束下,能同时保证算法的安全性和可达性.

## 2 问题描述与定义

本文讨论全局环境未知、且存在沿固定轨迹移动的动态障碍物的情况,其典型例子是同时存在自主车和沿导轨运动搬运小车的生产车间.规划的目的是使机器人由起点  $P_{int}$  安全避碰地到达终点  $P_{goal}$ .

记  $WS$  为机器人 Rob 在二维平面上的凸多边形有限运动区域,其内部分布着有限个静态障碍物  $SObs_1, SObs_2, \dots, SObs_n$ , 以及有限个动态障碍物  $DObs_1, DObs_2, \dots, DObs_m$ ; 将 Rob 模型化为点状机器人,同时  $WS$  中的静、动态障碍物已根据 Rob 的实际尺寸及安全性要求进行了相应“膨化”处理,并且使得“膨化”后的障碍物边界为安全区域,且各障碍物之间及障碍物与  $WS$  边界不相交; $DObs_i$  沿已知轨道单向运行, $t$  时刻  $DObs_i (i=1, \dots, m)$  的运动速率记为  $v_o^i(t), v_o^i(t) \in [0, v_{OH}]$ ,  $v_o^i(t)$  的变化不可预测.

Rob 无全局环境信息,在任一时刻,它只能实时探测到以其当前位置为中心, $r$  为半径区域内的环境信息(包括障碍物位置、速度);Rob 能向任意方向连续行走,运动时速率  $v_R$  恒定;环境探测及路径规划所需的时间忽略不计.

下面对机器人路径规划中的有关概念给出相应的定义.令  $S, D$  分别为静、动态障碍物的下标集.设  $WS$  中的点构成闭凸集  $W$ ,其边界为  $\partial W$ ;  $SObs_i$  中的点构成闭凸集  $SO_i$ ,其边界为  $\partial SO_i$ ;  $t$  时刻  $DObs_i$  中的点构成闭凸集  $DO_i(t)$ ,其边界为  $\partial DO_i(t)$ .

在  $WS$  中建立系统直角坐标系  $\Sigma_0$ ,则  $\forall P \in W$  在  $\Sigma_0$  都有确定的坐标  $(x, y)$ ;  $t$  时刻 Rob 的位置表示为  $P_R(t)$ ,其坐标为  $(x_R(t), y_R(t))$ ;规划起始时刻定为  $t=0$ .

**定义 1.**  $TO_i$  为障碍物的运动轨迹,指障碍物在  $WS$  中所能覆盖到的所有区域.对于静

态障碍物  $SObs_i, TO_i = SO_i, i \in S$ ; 对于动态障碍物  $DObs_j, TO_j = W \cap (\bigcup_{t=0}^{+\infty} DO_j(t)), j \in D$ .

显然, 应有

$$TO_i \cap TO_j = \Phi, \quad i \in S, j \in D \quad (1)$$

**定义 2.**  $d(P_i, P_j)$  为点  $P_i(x_i, y_i)$  到点  $P_j(x_j, y_j)$  的距离, 由公式

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

确定.

以下叙述中, 用上标“ $o$ ”, “ $c$ ”分别表示取相应集合的内部和补集.

**定义 3.**  $t$  时刻,  $\forall P \in W, P \in (SO_i^o \cup DO_j(t)^o), i \in S, j \in D$ , 则称  $P$  为  $t$  时刻的可行点,  $t$  时刻所有可行点的集合称为  $t$  时刻的可行域, 记为  $FD(t)$ ;  $t$  时刻,  $\forall P \in W$  或  $\forall P \in (SO_i^o \cup DO_j(t)^o), i \in S, j \in D$ , 则称  $P$  为  $t$  时刻的禁入点,  $t$  时刻所有禁入点的集合称为  $t$  时刻的禁入域, 记为  $NFD(t)$ .

显然

$$FD(t) = W \cap \left( \bigcap_{i \in S} (SO_i^o)^c \right) \cap \left( \bigcap_{j \in D} (DO_j(t)^o)^c \right) \quad (3)$$

$$NFD(t) = W^c \cup \left( \bigcup_{i \in S} SO_i^o \right) \cup \left( \bigcup_{j \in D} DO_j(t)^o \right) \quad (4)$$

特别地, 当  $D = \Phi$  时, 静态环境中的可行域记为  $FD$ .

**定义 4.**  $T = [t_0, t_f], \forall P_0 \in FD(t_0), \forall P_f \in FD(t_f)$ , 若连续映射  $FS(\cdot): T \rightarrow W$  使得  $FS(t_0) = P_0, FS(t_f) = P_f, FS(t) \in FD(t), t \in (t_0, t_f)$ , 则称映射  $FS(\cdot)$  是  $W$  中从  $P_0$  到  $P_f$  的一个可行路径.  $FS(t_0), FS(t_f)$  分别称为可行路径  $FS(\cdot)$  的起点和终点, 像集  $FS(T)$  称为从  $P_0$  到  $P_f$  的一条可行通道, 记为  $FP(P_0P_f)$ .

显然,  $FP(P_0P_f)$  构成了  $WS$  中连接点  $P_0(x_0, y_0), P_f(x_f, y_f)$  的连续曲线, 而可行路径  $FS(\cdot)$  则是指产生可行通道的连续映射; 并且对于同一可行通道, 可以有不同的可行路径.  $FP(P_0P_f)$  在  $WS$  中形成曲线的弧长称为路程, 记作  $FJ(P_0P_f)$ .

**定义 5.**  $\varphi(P_iP_j)$  表示  $WS$  中以  $P_i$  为起点,  $P_j$  为终点的一条连续曲线. 为表述方便,  $\varphi(P_iP_j)$  同时也表示该曲线上所有点的集合. 特别地, 当  $\varphi(P_iP_j)$  是直线时, 记为  $\bar{\varphi}(P_iP_j)$ .

**定义 6.**  $Win(P_R(t)) = \{P | P \in W, d(P, P_R(t)) \leq r\}$  称为 Rob 在点  $P_R(t)$  处的视野域, 亦即该点的滚动窗口, 其中  $P_R(t) \in FD(t)$ .

### 3 基于滚动窗口路径规划的基本原理及算法

上节提出的在信息不完全、环境不确定情况下的路径规划问题, 是传统全局规划方法无法解决的; 而依靠瞬时信息反馈的导航算法又因为缺乏对宏观环境的了解, 往往是低效率的, 并难以保证可达性. 注意到机器人在运动过程中能探知其传感范围内一有限区域的环境信息, 这部分信息必须充分利用, 因此解决这一问题的指导思想是: 采用反复进行的局部优化规划代替一次性的全局优化的结果, 并在每次局部优化规划中充分利用该时刻最新的局部环境信息. 文献[13]已把工业过程中应用成熟的预测控制原理推广到动态不确定环境下规划、调度、决策等问题的求解上, 提出了充分利用已知信息进行预测与评价, 通过滚动窗口进行局部优化, 并以反馈信息更新已知信息的三项原理, 可有效地应用到本文所讨论的机器

人路径规划问题中.

基于滚动窗口的路径规划算法的基本原理如下所述.

**场景预测:**在滚动的每一步,机器人根据其探测到的局部窗口范围内的环境信息,用启发式方法生成局部子目标,并对动态障碍物的运动进行预测,判断机器人行进是否可能与动态障碍物相碰撞.

**滚动窗口优化:**机器人根据窗口内的环境信息及预测结果,选择局部规划算法,确定向子目标行进的局部路径,并实施当前策略,即依所规划的局部路径行进一步,窗口相应向前滚动.

**反馈初始化:**在新的滚动窗口产生后,根据传感器所获取的最新信息,对窗口内的环境及障碍物运动状况进行更新.

以下将讨论实施以上滚动路径规划的几个关键问题,并在此基础上给出完整的算法.

### 3.1 局部子目标的确定

在基于滚动窗口的局部规划中,由于全局最终目标  $P_{\text{goal}}$  不一定包含在滚动窗口内,必须选定局部规划的子目标,它可看作  $P_{\text{goal}}$  在滚动窗口内的映射.局部子目标的产生方式为:在  $t$  时刻,Rob 的滚动窗口为  $\text{Win}(P_R(t))$ ,若  $P_{\text{goal}} \in \text{Win}(P_R(t))$ ,则取  $P_{\text{sub}}(t) = P_{\text{goal}}$ ;否则利用启发式函数  $f(P) = g(P) + h(P)$  来选取使  $f(P)$  最小的窗口边界点  $P$  作为子目标  $P_{\text{sub}}(t)$ ,即

$$\begin{aligned} \min_P f(P) &= g(P) + h(P) \\ \text{s. t. } P &\in \partial\text{Win}(P_R(t)) \end{aligned}$$

其中  $g(P)$  为 Rob 从当前位置行进到  $P$  的代价,其值可根据  $P$  的位置及当前窗口内的环境信息来估计, $h(P)$  为从  $P$  行进到终点的代价.由于边界外的信息无法获取,一般  $h(P)$  只能采用  $P$  到终点的距离来估计.这种子目标的选择方法反映了全局优化的要求与局部有限信息约束的折衷,是在给定信息环境下企图实现全局优化的自然选择.

本文为简化计算,降低对优化的要求,着重研究寻找可行的算法,为此仅以  $P$  是否属于可行域来规定  $g(P)$  的值,即

$$g(P) = \begin{cases} 0, & \text{if } (P \in FD(t)), \\ +\infty, & \text{if } (P \in NFD(t)). \end{cases}$$

这样,子目标的选择问题可归结为如下优化问题

$$\begin{aligned} \min J &= \min_P d(P, P_{\text{goal}}) \\ \text{s. t. } P &\in \partial\text{Win}(P_R(t)) \cap FD(t) \end{aligned} \quad (5)$$

### 3.2 机器人与动态障碍物相碰的预测

令  $DW(t) = \{i | DO_i^o(t) \cap \text{Win}(P_R(t)) \neq \Phi\}$ ,表示  $t$  时刻滚动窗口内可探测到的动态障碍物的下标集.

根据对环境的约定, $t$  时刻 Rob 能够对窗口内探测到的动态障碍物  $DObs_i (i \in DW(t))$  今后的运动路线作出准确预测.但由于  $DObs_i$  的运动速率不定,Rob 无法准确预知其在将来某时刻的位置,而只能按其最大运行速率对未来某一时段内  $DObs_i$  可能出现的最大范围作出预测.假定在某一时刻  $t$  按原计划 Rob 将笔直朝目标  $P_f$  行进,则该行进过程根据  $\bar{\varphi}(P_R(t)P_f)$  和  $DObs_i$  的相对位置,可分解为以下五种基本情况(见图 1)的组合,即

- 1)  $P_R(t) \notin TO_i^o, P_f \notin TO_i^o$ , 且  $\bar{\varphi}(P_R(t)P_f) \cap TO_i^o = \Phi$ ;
- 2)  $P_R(t) \in TO_i^o, P_f \in \partial TO_i$ , 且  $\bar{\varphi}(P_R(t)P_f) \subset TO_i$ ;
- 3)  $P_R(t) \in TO_i^o, P_f \in TO_i^o$ , 且  $\bar{\varphi}(P_R(t)P_f) \subset TO_i^o$ ;
- 4)  $P_R(t) \in \partial TO_i, P_f \in \partial TO_i$ , 且  $\bar{\varphi}(P_R(t)P_f) \subset TO_i$ ;
- 5)  $P_R(t) \in \partial TO_i, P_f \in TO_i^o$ , 且  $\bar{\varphi}(P_R(t)P_f) \subset TO_i$ .

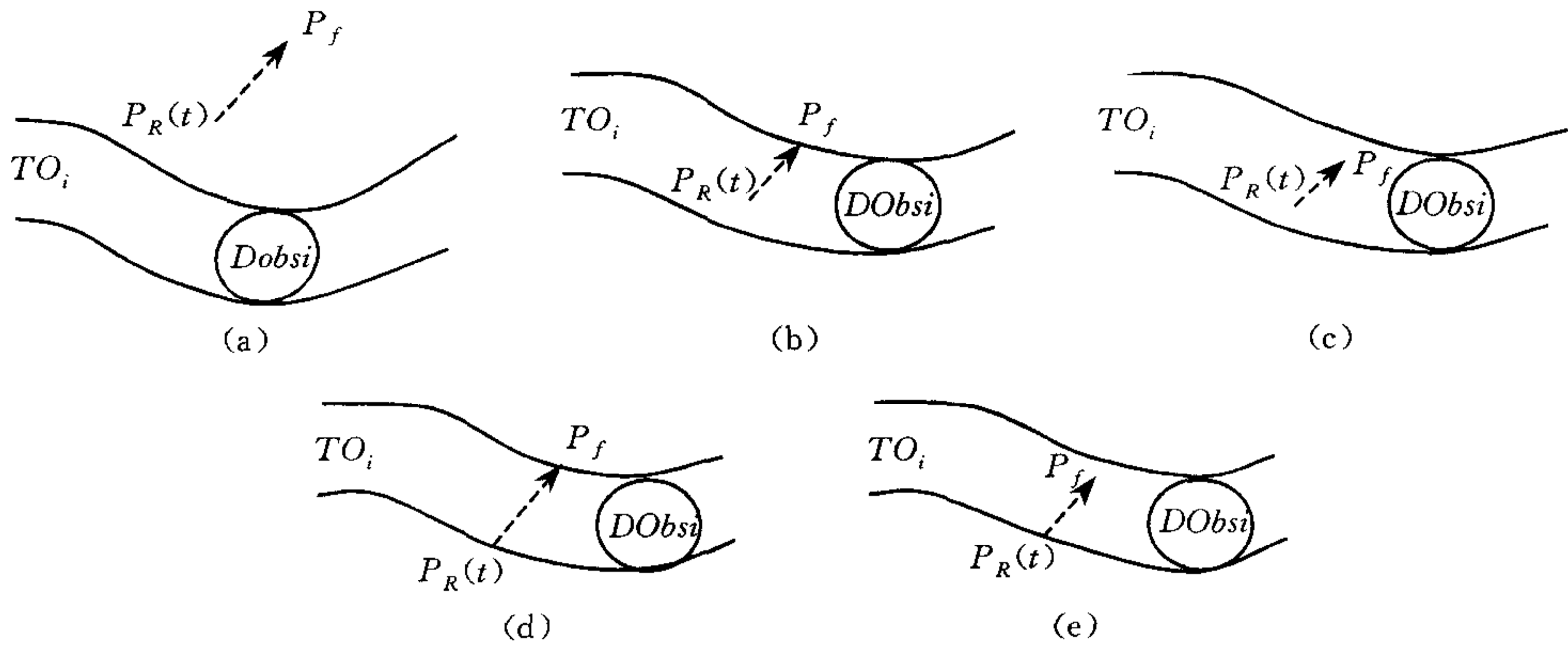


图 1 Rob 与 DObsi 的相对位置情况 ( $i \in DW(t)$ )

对于情况 1), DObsi 的出现不影响 Rob 向目标前进, 可不予考虑, 称为直接可逃逸.

对于情况 2), Rob 在 DObsi 运动轨迹内部, 若按原计划, Rob 将走到障碍物轨道边界  $\partial TO_i$  的  $P_f$  点. 但这样可能与 DObsi 相碰, 所以 Rob 需对未来的场景作一预测, 以判断是否可能与动态障碍碰撞. Rob 从  $P_R(t)$  运行至  $P_f$  需时  $t_p = \frac{d(P_R(t), P_f)}{v_R}$ , 因此就以  $t_p$  为预测时域(下同). 取  $v_o^i(\tau) = v_{OH}, \tau \in [t, t+t_p]$ , 因为障碍物运行轨迹已知, 故可求得 DObsi 在当前滚动窗口内  $t_p$  时间段里所能覆盖的最大区域

$$TO_i|_t^{t+t_p} = \text{Win}(P_R(t)) \cap \left( \bigcup_{\tau=t}^{t+t_p} \tilde{D}\tilde{O}_i(\tau) \right) \tag{6}$$

其中  $\tilde{D}\tilde{O}_i(\tau)$  为预测值, 是按 DObsi 以最大速率  $v_{OH}$  沿已知轨道运行计算得到的.

为简化计算及安全性考虑, 作以下碰撞判断: 若  $TO_i|_t^{t+t_p} \cap \bar{\varphi}(P_R(t)P_f) = \Phi$  (见图 2(a)), 则认为 Rob 按原计划行进不会与 DObsi 碰撞, 亦称为直接可逃逸; 若  $TO_i|_t^{t+t_p} \cap \bar{\varphi}(P_R(t)P_f) \neq \Phi$ , 即认为 Rob 立刻沿直线向  $P_f$  运动, 有可能产生碰撞, 其中如从  $t$  时刻起 DObsi 正沿轨道远离  $P_R(t)$  (见图 2(b)), 称为间接可逃逸(此时 Rob 不需改变运动路线, 只要原地作适当等待); 如从  $t$  时刻起 DObsi 正沿轨道靠近  $P_R(t)$  (见图 2(c)), 称为不可逃逸.

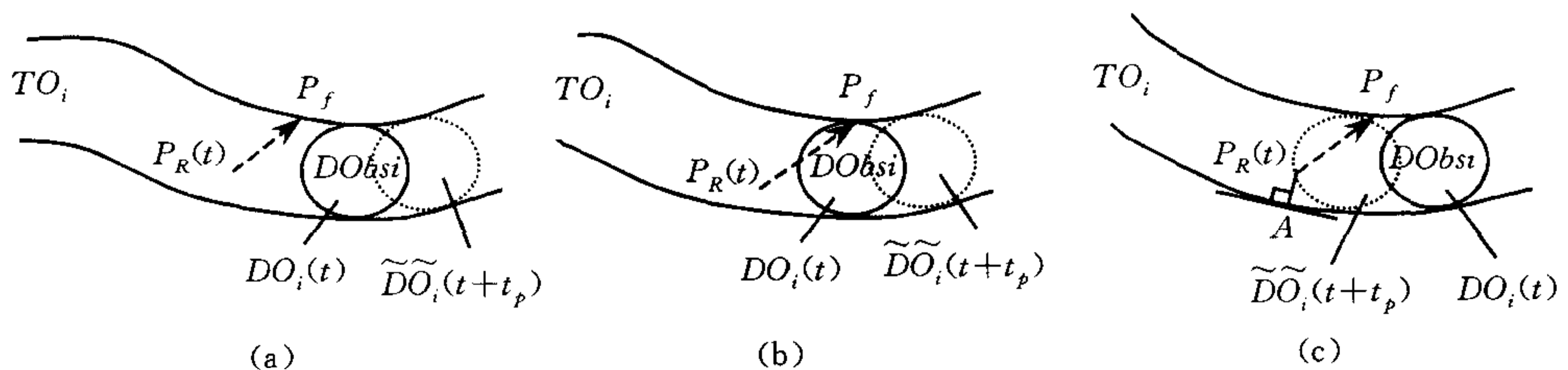


图 2 情况 2) 中的碰撞预测

对于情况 3), 需要作与情况 2) 中相似的预测处理, 不过由于终点  $P_f$  在动态障碍物运行轨道内部, 故当  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) = \Phi$ , 且从  $t$  时刻起  $Dobsi$  正沿轨道靠近  $P_R(t)$  时仍作为不可逃逸; 除此之外, 碰撞判断同情况 2).

对于情况 4), 若  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) = \Phi$ , 为直接可逃逸; 若  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) \neq \Phi$ , 为间接可逃逸.

对于情况 5), 若  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) = \Phi$ , 且从  $t$  时刻起  $Dobsi$  正沿轨道远离  $P_R(t)$ , 为直接可逃逸; 若  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) = \Phi$  且从  $t$  时刻起  $Dobsi$  正沿轨道靠近  $P_R(t)$ , 或  $TO_i|_i^{t+\Delta t} \cap \bar{\varphi}(P_R(t)P_f) \neq \Phi$ , 为间接可逃逸.

在以上的分析中, 必需注意两点: 一是 Rob 若在动态障碍轨道外或轨道边界上, 则是安全的; 二是直接可逃逸是指在预测全过程及其后 Rob 按计划行进都不会与  $Dobsi$  碰撞, 间接可逃逸是指 Rob 目前所处位置在预测全过程及其后都不会与  $Dobsi$  碰撞, 而不可逃逸则是指 Rob 目前或将要行进到的位置均有可能与  $Dobsi$  碰撞.

### 3.3 局部规划算法

在对机器人与动态障碍物相碰作出预测的基础上, 为避免相碰, 可采取下述策略.

对于情况 2), 当直接可逃逸时, Rob 立即按原规划路径运行; 当间接可逃逸时, Rob 只需在原地等待, 直至朝终点笔直行走不再会与该运动障碍物碰撞, 即可转化为直接可逃逸时, 可依规划路线行进; 当不可逃逸时, Rob 应放弃原行进计划, 而采取必要的避碰措施, 本文中使用时最短时间策略, 即 Rob 沿  $Win(P_R(t))$  内能最快到达  $\partial TO_i$  的路线运行, 如图 2c) 中的直线段  $\bar{\varphi}(P_R(t)A)$ , 并在  $A$  点等待到可转化为直接可逃逸状态再回复到  $P_R(t)$ , 此后再考虑朝终点行进, 其安全性将在下节进一步分析.

对于情况 3)~情况 5) 中的直接、间接可逃逸和不可逃逸状态均可采取以上相同措施. 由于实际情况是上述基本情况的组合, 故具体实施规划时应参照各基本情况分段考虑.

采取以上避碰策略并结合局部窗口范围内已知的静态障碍信息, 可给出一个 Rob 在动态可探知环境中由某一起始点  $P_0$  到某一目标点  $P_f$  的可行路径规划算法, 其中  $P_R(t_0) = P_0$ ,  $P_0 \in FD(t_0)$ ,  $P_f \in SO_j^0, j \in S$ .

**算法 1.** 动态可探知环境中的停-等算法.

Step1. 如果  $P_R(t) \in TO_i^0$ , 转 Step3; 否则, 转 Step2.

Step2. I. 若直接可逃逸, 朝终点笔直行走, 直至以下任一情况发生:

1) 终点已达到, 规划结束;

2) 走到该动态障碍物运动轨迹接近终点  $P_f$  的边界, 转 Step1.

II. 若间接可逃逸, 等待直至朝终点笔直行走而不再会与该运动障碍物碰撞, 转 I.

III. 若不可逃逸, 以最短时间策略行进到该动态障碍物运动轨迹边界, 等待直至  $DObSi$  完全通过  $P_R(t)$ , 立即按原路返回  $P_R(t)$ , 转 I.

Step3. 朝终点笔直行走, 直至以下任一情况发生:

1) 终点已达到, 规划结束;

2) 走到某静态障碍物边界, 转 Step4;

3) 走到某动态障碍物运动轨迹边界并且即将进入该轨迹内部时, 转 Step2.

Step4. 沿能使到终点距离减小的该障碍物一侧边界行走, 直至以下任一情况发生:

- 1) 终点已达到, 规划结束;
- 2) 朝终点笔直行走而不会碰到该障碍物, 转 Step3.

算法 1 中 Rob 需对探测到的动态障碍物今后的运动状况进行预测, 确定可能会与自己碰撞的动态障碍物, 在行走中不断监视上述动态障碍物的运动, 并根据当前自身位置情况采取相应措施. 其中 Step2 中 II, III 对应图 1(b)~图 1(e)所示在动态障碍物轨迹区内的行走策略; Step3 对应在动态障碍物轨迹区外的行走策略, 同时它还考虑了遇到静态障碍物及新的动态障碍物的可能性; 而 Step4 则是在静态障碍环境中的行走策略.

### 3.4 基于滚动窗口的路径规划算法

在解决了上述关键问题后, 可以得到以下基于滚动窗口的路径规划算法, 其中  $P_{\text{int}} \in FD(0)$ ,  $P_{\text{goal}} \in SO_i$ ,  $i \in S$ .

**算法 2.** 基于滚动窗口的路径规划算法.

Step0. 对起点 ( $P_{\text{int}}$ )、终点 ( $P_{\text{goal}}$ )、工作环境 ( $WS$ )、Rob 的视野半径 ( $r$ )、步长 ( $\epsilon$ ) 进行初始化;

Step1. 如果终点到达, 规划中止;

Step2. 对当前滚动窗口  $Win(P_R(t))$  内的环境信息进行更新;

Step3. 产生局部子目标  $P_{\text{sub}}(t)$ ;

Step4. 采用算法 1 进行窗口内的局部路径规划, 并依局部规划策略行进, 步长为  $\epsilon$  ( $0 < \epsilon < r$ );

Step5. 返回 Step1.

本算法中 Step3 根据 (5) 式产生局部子目标时只须考虑当前滚动窗口内探测到的所有静态障碍物, 而不必考虑动态障碍物, 即选择  $t$  时刻窗口边界上不为静态障碍物所覆盖且离  $P_{\text{goal}}$  最近的点作为局部子目标  $P_{\text{sub}}(t)$  ( $P_{\text{sub}}(t) \in FD$ ), 易证此时子目标总是存在的<sup>[14]</sup>. 另外, Step4 的行进步长  $\epsilon$  不包括算法 1 中 Rob 因不可逃逸而执行避障措施 Step2 中的 III 而走过的路程.

## 4 算法安全性及可达性分析

基于滚动窗口的路径规划使机器人对动态未知环境具有良好的适应性, 但由于缺乏全局环境信息, 上述滚动规划算法并不能保证机器人在任意复杂的环境下都能安全地到达终点. 下面来讨论动态未知环境下, 满足什么条件, 可以保证上述滚动规划算法的安全性和可达性.

对  $WS$  附加以下约定:

A1) 假定所有静态障碍物为凸型, 边界为光滑曲线, 且边界上各点曲率半径  $\leq \delta < r$  (其中  $\delta$  是正常量), 曲率中心在障碍物内部; 动态障碍物为圆形, 且半径  $< \delta$ ;

A2)  $DObs_i$  沿一不闭合, 不自相交的光滑平缓轨道 (在任一滚动窗口内的部分都可近似为直线段) 单向行进, 在  $WS$  内的平均速率  $\bar{v}_i \geq \bar{v}_0$ , 其中  $\bar{v}_0$  为大于零的常量;

A3) 各动态障碍物之间的最近距离  $> 2r$ .

此外, Rob 初始位置  $P_R(0)$  与任一移动障碍物之间的距离满足

$$d(P_R(0), R) \geq r - \epsilon - v_{OH} \frac{\epsilon}{v_R}, \quad \forall R \in DO_i, i \in D \tag{7}$$

为论述方便,先给出有关定义.

**定义 7.** 设可行路径  $FS^*$ , 满足  $FS^*(t_0) = P_0, FS^*(t_f) = P_f$ , 若  $\forall t_1, t_2 \in [t_0, t_f], t_1 < t_2, FS^*(t_1) \neq FS^*(t_2)$ , 都有  $d(FS^*(t_2), P_f) < d(FS^*(t_1), P_f)$ , 则称  $FS^*$  为从  $P_0$  到  $P_f$  的严格趋近可行路径.

**定义 8.** 令  $d_i$  为  $DObs_i$  的直径. 称  $L$  为动态障碍物的最大尺寸,  $L = \max_{i \in D} \{d_i\}$ ;  $F$  为动态障碍物的最小尺寸,  $F = \min_{i \in D} \{d_i\}$ .

**定义 9.** 若存在  $t_g \geq 0$ , 使  $P_R(t_g) \in \{P | P \in W, D(P, P_{goal}) < \epsilon\}$ , 则称 Rob 于  $t_g$  到达终点  $P_{goal}$ .

### 4.1 安全性分析

**引理 1.**  $\forall t \geq 0$ , Rob 在  $Win(P_R(t))$  内最多只能探测到一个动态障碍物.

由环境约束 A3) 有

$$d(P_1, P_2) > 2r, \quad \forall P_1 \in DO_i(t), \quad \forall P_2 \in DO_j(t) \tag{8}$$

其中  $t \geq 0; i, j \in D, i \neq j$ . 易知引理成立.

因此,在本次窗口内进行局部规划时,Rob 最多只需监视一个动态障碍物的运动情况.

用  $(t + \Delta t)$  表示 Rob 以固定步长行进一步后新的时刻(由于实施局部规划策略时可能会产生等待,所需时间不定,故  $\Delta t$  为变量); 并令

$$Q_i(t) \in \{P | d(P_R(t), P) = \min d(P_R(t), Q), \quad P, Q \in DO_i(t)\},$$

它是  $t$  时刻动态障碍物  $DObs_i$  上距 Rob 最近的点.

**引理 2.** 在滚动规划中,若  $t(t \geq 0)$  时刻第  $k(k \geq 1)$  次局部规划的实施能使 Rob 与  $DObs_i(i \in DW(t))$  避碰, 则第  $(k + 1)$  次的局部规划必能使 Rob 与  $DObs_i$  避碰.

**证明.** 当第  $k$  次局部规划的实施结束, 即 Rob 按  $k$  次规划行进  $\epsilon$  后, 有  $P_R(t + \Delta t) \notin TO_i^c$ , 表明 Rob 原本不在或已脱离  $TO_i^c$ , 则显然第  $(k + 1)$  次的局部规划必能使 Rob 与  $DObs_i$  避碰. 当  $P_R(t + \Delta t) \in TO_i^c$  时, 考虑 Rob 与  $DObs_i(i \in DW(t))$  可能的各种基本情况, 若第  $k$  次局部规划时, 存在间接可逃逸或不可逃逸路段, Rob 通过采取 3.3 节所述相应策略达到避碰, 并且 Rob 行进  $\epsilon$  后停留在上述路段, 则由环境约定和规划策略知, 此时  $DObs_i$  已远离  $P_R(t + \Delta t)$ , 故第  $(k + 1)$  次的局部规划必能产生使 Rob 与  $DObs_i$  避碰的可行路径. 若第  $k$  次

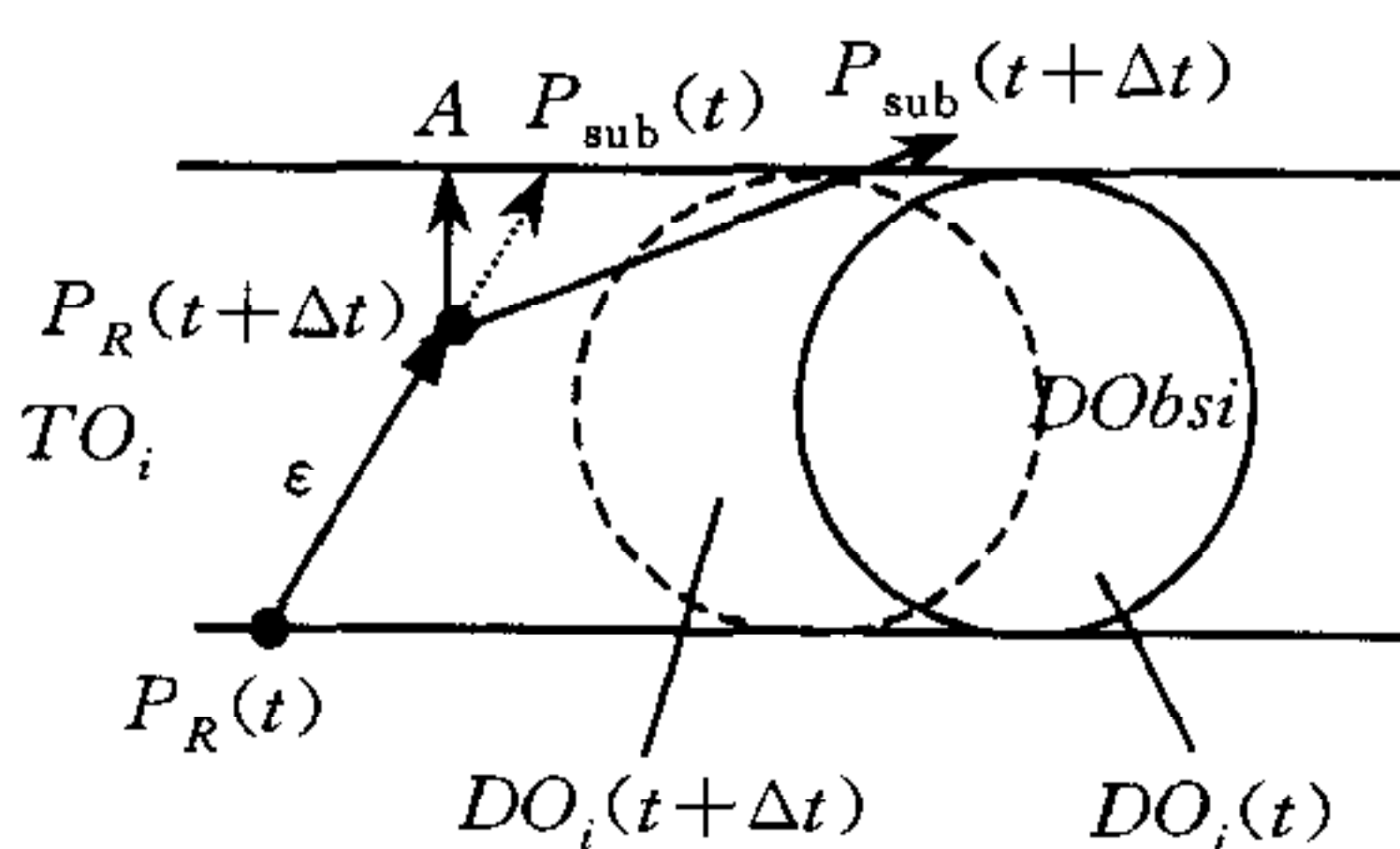


图 3 引理 2 的证明

局部规划时, 存在直接可逃逸路段  $\varphi(P_R(t), P_{sub}(t))$ , 并且 Rob 行进  $\epsilon$  后停留在上述路段, 这时 Rob 仍旧根据当前滚动窗口内  $DObs_i$  的运行情况判断自己是处于直接可逃逸、间接可逃逸或是不可逃逸状态, 若是前两种情况, 显然第  $(k + 1)$  次的局部规划必能使 Rob 与  $DObs_i$  避碰; 若是不可逃逸, 如图 3 所示, 由算法 1 知应采取最短时间策略行进到动态障碍物轨道边界的  $A$  点, 因为

$$d(P_R(t + \Delta t)A) \leq d(P_R(t + \Delta t)P_{sub}(t)),$$

且  $DObs_i$  为圆形, 易证 Rob 可以速度  $v_R$  安全到达  $A$  点, 这样本次规划中 Rob 必能避免与

$Dobsi$  的碰撞. 综上所述, 若  $t$  时刻第  $k$  次局部规划的实施能使 Rob 与  $Dobsi (i \in DW(t))$  避碰, 则第  $(k+1)$  次的局部规划必能使 Rob 与  $Dobsi$  避碰. 证毕.

引理 2 表明, 对于滚动窗口内的某一已探知动态障碍物, 若 Rob 在第  $k$  次局部规划中能与之避碰, 则第  $(k+1)$  次的局部规划同样能保证与之避碰.

**引理 3.** 在滚动规划中, 当  $t (t \geq 0)$  时刻进行第  $k (k \geq 1)$  次局部规划时  $DO_i(t)^\circ \cap \text{Win}(P_R(t)) = \Phi$ , Rob 依局部规划顺利行进一步, 而第  $(k+1)$  次局部规划时  $i \in DW(t+\Delta t)$ , 若有

$$\frac{v_R}{v_{OH}} \geq \frac{\frac{L}{2} + \epsilon}{r - \epsilon} \quad (9)$$

则第  $(k+1)$  次的局部规划必能使 Rob 与  $Dobsi$  避碰.

**证明.** 由引理 1 知,  $\text{Win}(P_R(t+\Delta t))$  内只可能有一个动态障碍物, 即  $Dobsi$ .  $t$  时刻在 Rob 即将进行第  $k (k \geq 1)$  次规划时, 动态障碍物  $Dobsi$  未能被 Rob 探测到, 即  $d(P_R(t), Q_i(t)) \geq r$ ; 若  $Dobsi$  沿  $\bar{\varphi}(Q_i(t)P_R(t))$  以最大速率  $v_{OH}$  向  $P_R(t)$  运行, 而局部规划的实施又恰使 Rob 沿  $\bar{\varphi}(P_R(t)Q_i(t))$  以速率  $v_R$  向  $Q_i(t)$  运行, 步长为  $\epsilon$ . 这样, 任何  $t$  时刻未知, 但  $(t+\Delta t)$  时刻被探知的动态障碍物离 Rob 的最近距离  $d(P_R(t+\Delta t), Q_i(t+\Delta t)) \geq \left( r - \epsilon - v_{OH} \times \frac{\epsilon}{v_R} \right)$ . 若要有效避碰, 不仅当前局部规划策略实施完以前不能碰撞, 而且在第  $(k+1)$  次局部规划时要使机器人有足够的时间完成不可逃逸时的避碰策略, 即实施算法 1 的 Step2 中的 III, 其所需时间不大于  $(L/2)/v_R$ . 由引理 1, 在  $(t+\Delta t)$  时刻的规划只需考虑  $Dobsi$  的影响, 又由式(1)易知 Rob 实施算法 1 的 Step2 中的 III 时是在  $\text{Win}(P_R(t))$  内的可行域中移动, 由图 3 可知, 若其速率  $v_R$ , 步长  $\epsilon$  及视野半径  $r$  满足

$$\frac{v_R}{v_{OH}} \geq \frac{\frac{L}{2} + \epsilon}{r - \epsilon},$$

则有

$$\frac{L}{2} \leq \frac{r - \epsilon - v_{OH} \times \frac{\epsilon}{v_R}}{v_{OH}},$$

即 Rob 逃逸至边界  $A$  点的时间小于  $Dobsi$  移至  $P_R(t+\Delta t)$  的时间, 因此必能顺利执行算法 1 的避障措施 Step2 中的 III. 证毕.

引理 3 表明, 对于  $t$  时刻第  $k$  次局部规划时未探测到, 而  $(t+\Delta t)$  时刻第  $(k+1)$  次局部规划时才探测到的动态障碍物, 若满足(9)式, 则在第  $(k+1)$  次局部规划中 Rob 必能与之避碰.

**推论 1.** 在当前滚动窗口内用算法 1 进行局部规划时, 若  $P_R(t) \in TO_i^\circ, i \in DW(t)$ , 但 Rob 不可逃逸, 并且 Step2 中 III 的避碰措施得以实施, 若满足式(9), 则当时刻  $\tau (\tau > t)$  Rob 再次返回  $P_R(t)$  时, 有  $DO_i(\tau) \cap \text{Win}(P_R(\tau)) \neq \Phi$ , 即  $DObsi$  尚未完全离开当前滚动窗口.

**证明.** 由算法 1, 显然有  $P_R(t) = P_R(\tau), \text{Win}(P_R(t)) = \text{Win}(P_R(\tau))$ . 因为 Rob 不可逃逸, 故  $Dobsi$  必经过点  $P_R(t)$ .

由式(9)知

$$\begin{aligned} \frac{v_R}{v_{OH}} &\geq \frac{\frac{L}{2} + \epsilon}{r - \epsilon} > \frac{L}{2r}, \\ &\Rightarrow \frac{r}{v_{OH}} > \frac{L}{2v_R} \end{aligned} \quad (10)$$

上式中左端为  $DObsi$  完全走出滚动窗口  $Win(P_R(t))$  所需的最短时间, 而右端为 Rob 返回  $P_R(t)$  所需的最长时间, 因此有  $DO_i(\tau) \cap Win(P_R(\tau)) \neq \Phi$ . 证毕.

**定理 1.** 在滚动规划中, 若满足式(9), 则  $t(t \geq 0)$  时刻开始的第  $k(k \geq 1)$  次局部规划中, Rob 如能与  $Dobsi(i \in DW(t))$  避碰, 则本次局部规划必能顺利实施 (Rob 依局部规划安全地走完一步).

**证明.** 若  $t$  时刻 Rob 直接或间接可逃逸, 则 Rob 必能顺利实施本次局部规划; 若  $t$  时刻 Rob 不可逃逸, 实施算法 1 的 Step2 中的 III 后, 由推论 1 知, 当 Rob 返回  $P_R(t)$  时, 所要避碰的动态障碍物  $DObsi$  尚未离开滚动窗口; 再由环境约束 A2) 及引理 1, 当前只需考虑  $DObsi$  的影响, 且本次局部规划结束前 Step2 中的 III 不会再被执行, 故此 Rob 必能顺利实施本次局部规划. 证毕.

**定理 2.** Rob 在满足环境约定的任何情况下实施滚动规划, 若有

$$\frac{v_R}{v_{OH}} \geq \frac{\frac{L}{2} + \epsilon}{r - \epsilon},$$

则始终能保证 Rob 安全行进.

**证明.** 1) 当进行第一次滚动规划时

若  $DW(0) = \Phi$ , 显然 Rob 能依局部规划的可行路径安全走完一步.

若  $DW(0) \neq \Phi$ , 由式(7)知, Rob 必能与滚动窗口内的动态障碍物避碰. 再由定理 1, Rob 必能安全行进一步.

2) 如  $t$  时刻开始的第  $k(k \geq 1)$  次滚动规划顺利结束 (Rob 安全走完第  $k$  步), 在  $(t + \Delta t)$  时刻

若  $DW(t + \Delta t) = \Phi$ , 则第  $(k + 1)$  次滚动规划必能顺利实施.

若  $DW(t + \Delta t) \neq \Phi$ , 如果  $DW(t) = \Phi$ , 由引理 1, 滚动窗口内仅有一个动态障碍物, 再由引理 3 知 Rob 必能与滚动窗口内的动态障碍物避碰; 如果  $DW(t) \neq \Phi$ , 且  $i = j$ , 其中  $i \in DW(t)$ ,  $j \in DW(t + \Delta t)$ , 则由引理 2 知 Rob 必能与滚动窗口内的动态障碍物避碰; 如果  $DW(t) \neq \Phi$ , 且  $i \neq j$ , 其中  $i \in DW(t)$ ,  $j \in DW(t + \Delta t)$ , 则由引理 3 知 Rob 必能与滚动窗口内的动态障碍物避碰. 再由定理 1, 本次规划必能顺利实施, 即 Rob 安全行进一步.

由以上分析归纳可知, 对于任意  $t \geq 0$ , 只要 Rob 满足式(9), 必能保证滚动规划算法 2 的安全性. 证毕.

由定理 2 可以看出:

a) 当  $\epsilon \geq r$  时, 不能保证避碰;

b) 对 Rob 而言,  $v_R$  越大,  $r$  越大,  $\epsilon$  越小, 就越有利于避碰;

c) 只要有较大的视野范围  $\left(r > \frac{L}{2} + 2\epsilon\right)$ , Rob 的行进速率完全可以小于动态障碍物的速率.

以下讨论中, 皆以满足式(9)为前提.

## 4.2 可达性分析

文献[14]已对静态障碍环境下机器人滚动路径规划算法的可达性作了分析,在某些情况下动态环境下规划路径对应的通道与静态环境下具有一致性. 本节将针对环境中存在移动障碍物的新特点,着重证明 Rob 依算法 1 所规划的路径行进能向子目标趋近,它也不会 在所规划的路径上无限等待,并且当存在移动障碍物时,滚动规划的子目标能不断向终点严格趋近,在此基础上最终借用文献[14]中的部分结论证明本文规划算法的可达性.

以下简称当  $P_R(t) \in TO_i^o$ ,  $i \in DW(t)$ , Rob 可逃逸(包括直接、间接可逃逸)的情况为内点可逃逸;反之,称为内点不可逃逸.

**引理 4.** 若规划中内点可逃逸,则对于有相同静态障碍物的可探知环境,不管动态障碍物如何分布、运行,从  $t$  时刻用算法 1 规划出的可行路径所对应的可行通道都是完全一样的;并且,这时规划出的路径是从起始点  $P_0$  到目标点  $P_f$  的严格趋近路径.

**证明.** 由算法 1 的 Step2 中的 II 知,动态障碍物只影响 Rob 在某处的等待时间,而对今后运行的方向丝毫没有影响,故对应可行通道就是静态环境中规划所得的可行通道,它是唯一确定的;并且由环境约束 A1) 易证,它是从  $P_0$  到  $P_f$  的严格趋近路径<sup>[14]</sup>.

对于确定的可行通道  $FP(P_0P_f)$ ,  $FJ(P_0P_f)$  是有限的,而  $DObs_i$  的最小尺寸  $F > 0$ ,故由环境约束 A2 易知,某一动态障碍物  $DObs_i$  只可能有限次通过该通道,且通过的最多次数  $N = \left\lceil \frac{FJ(P_0P_f)}{F} + 1 \right\rceil$ . 证毕.

**定理 3.** 当动态可探知环境中只有一个动态障碍物  $DObs_i$  时,算法 1 一定能保证 Rob 在有限时间内由  $P_0$  安全到达  $P_f$ .

**证明.** 因为只有一个动态障碍物  $DObs_i$ ,再由环境约束 A2)、式(7)及算法 1 步骤,可知 Step2 中的 II 或 III 只可能在规划刚开始时被执行一次;而且由式(1)知,执行 III 时,Rob 是在可行域中移动. 设执行 II 或 III 需时  $T_0$  (显然为有限值),它的执行结果等同于将原规划问题转化为从  $t = T_0$  时刻开始,在定理要求环境中规划一条从  $P_0$  ( $P_0 \in FD(T_0)$ , 且  $P_0 \notin TO_i^o$ ) 到  $P_f$  的可行路径,所以 II 或 III 的实施并不会影响算法的可达性. 考虑 II 或 III 执行后的情况,由引理 4 知,以后规划出的可行路径所对应通道同静态环境中,它是由静态障碍环境唯一确定的. 而静态环境中,Rob 显然能在有限时间内由  $P_0$  安全到达  $P_f$ ,所需时间为  $T = \frac{FJ(P_0P_f)}{v_R}$ . 另外,为确保不与  $DObs_i$  碰撞,Rob 最多需等待  $N = \left\lceil \frac{FJ(P_0P_f)}{F} + 1 \right\rceil$  次,由于  $\bar{v}_o \geq \bar{v}_o > 0$ ,故总等待时间  $T_w$  为有限值. 所以 Rob 总能在有限时间  $(T_0 + T + T_w)$  内由  $P_0$  安全到达  $P_f$ . 证毕.

用  $LFS$  表示滚动窗口内从  $P_R(t)$  到  $P_{sub}(t)$  的局部可行路径,其中  $LFS(t_0) = P_R(t)$ ,  $LFS(t_f) \in \partial Win(P_R(t))$ ,  $LFS(\tau) \in Win(P_R(t))$ ,  $\tau \in (t_0, t_f)$ ; 对应通道记为  $LFP(P_R(t), P_{sub}(t))$ .

**引理 5.** 在每次滚动规划时,若滚动窗口中内点可逃逸,则一定能在  $Win(P_R(t))$  中规划出可行路径  $LFS$ ,使得  $\forall \tau_1, \tau_2 \in [t_0, t_f], \tau_1 < \tau_2$ ,  $LFS(\tau_1) \neq LFS(\tau_2)$  时,都有  $d(LFS(\tau_2), P_{sub}(t)) < d(LFS(\tau_1), P_{sub}(t))$ . 称这样的  $LFS$  是  $Win(P_R(t))$  内由  $P_R(t)$  向  $P_{sub}(t)$  严格趋近的可行路径.

**证明.** 由引理 1, 4 及定理 3 知一定存在由  $P_R(t)$  向  $P_{sub}(t)$  严格趋近的可行路径,从而易

证引理 5 成立.

证毕.

**引理 6.** 在约定环境中使用算法 2, 每次局部规划实施结束都有  $d(P_R(t+\Delta t), P_{\text{sub}}(t)) < d(P_R(t), P_{\text{sub}}(t))$ .

**证明.** 在当前滚动窗口中用算法 1 进行局部规划, 若内点可逃逸, 则由引理 5 易知必能在当前窗口内规划出一条向  $P_{\text{sub}}(t)$  严格趋近的路径, 实施算法 2 的 Step4 后, 即有  $d(P_R(t+\Delta t), P_{\text{sub}}(t)) < d(P_R(t), P_{\text{sub}}(t))$ . 若内点不可逃逸, 首先执行算法 1 的 Step2 中的 III, 由推论 1 知, 当 Rob 返回  $P_R(t)$  时, 所要避碰的动态障碍物  $DObs_i$  尚未离开滚动窗口, 但已转化为直接可逃逸; 再由环境约束 A2) 及引理 1, 当前只需考虑  $DObs_i$  的影响, 且本次局部规划结束前 Step2 中的 III 不会再被执行, 算法 1 中其余各步将顺利实施, 由引理 5 此后必能在当前窗口内规划出一条从  $P_R(t)$  向  $P_{\text{sub}}(t)$  严格趋近的可行路径. 又算法 2 的 Step4 是指局部规划算法 1 的 Step3 被执行后开始的一步, 故在规划实施结束后总有  $d(P_R(t+\Delta t), P_{\text{sub}}(t)) < d(P_R(t), P_{\text{sub}}(t))$ .

证毕.

引理 6 表明, 在每次实施算法 2 的 Step4 时, 不管在执行算法 1 时遇到什么情况, Rob 每步的行进结果均使其向  $P_{\text{sub}}(t)$  严格趋近.

**引理 7.** 当  $t(t \geq 0)$  时刻进行第  $k(k \geq 1)$  次局部规划时, 若  $P_{\text{goal}} \in \text{Win}(P_R(t))$ , 点  $V \in \text{Win}(P_R(t))^\circ$ , 且  $V \in FD$ , 则必有

$$d(P_{\text{sub}}(t), P_{\text{goal}}) < d(V, P_{\text{goal}}).$$

**证明.** 因为静态环境中存在从  $V$  到  $P_{\text{goal}}$  的严格趋近可行路径  $FS^*$ , 对应通道记为  $FP^*(VP_{\text{goal}})$ . 又  $P_{\text{goal}} \in \text{Win}(P_R(t))$ , 所以

$$FP^*(VP_{\text{goal}}) \cap \partial \text{Win}(P_R(t)) \neq \Phi.$$

令  $P^* \in FP^*(VP_{\text{goal}}) \cap \partial \text{Win}(P_R(t))$ , 则  $d(P^*, P_{\text{goal}}) < d(V, P_{\text{goal}})$ . 根据子目标的选择算法,  $P_{\text{sub}}(t)$  必满足  $d(P_{\text{sub}}(t), P_{\text{goal}}) \leq d(P^*, P_{\text{goal}})$ , 因此有

$$d(P_{\text{sub}}(t), P_{\text{goal}}) < d(V, P_{\text{goal}}).$$

证毕.

**引理 8.** 用算法 2 进行路径规划, 若  $P_{\text{goal}} \in \text{Win}(P_R(t))$ , 则  $P_{\text{sub}}(t)$  是向  $P_{\text{goal}}$  严格趋近的, 即  $\forall t_1, t_2 \geq 0, t_1 < t_2$ , 有

$$d(P_{\text{sub}}(t_2), P_{\text{goal}}) < d(P_{\text{sub}}(t_1), P_{\text{goal}}),$$

其中  $t_1, t_2$  为任两次局部规划的起始时刻.

**证明.** 由引理 6 知每次局部规划实施结束都有

$$d(P_R(t+\Delta t), P_{\text{sub}}(t)) < d(P_R(t), P_{\text{sub}}(t)).$$

又  $d(P_{\text{sub}}(t+\Delta t), P_R(t+\Delta t)) = d(P_{\text{sub}}(t), P_R(t)) = r$ , 故  $P_{\text{sub}}(t) \in \text{Win}(P_R(t+\Delta t))^\circ$ . 由引理 7 知, 必有  $d(P_{\text{sub}}(t+\Delta t), P_{\text{goal}}) < d(P_{\text{sub}}(t), P_{\text{goal}})$ , 再由  $t$  的任意性, 易知引理成立. 证毕.

引理 8 表明, 连续的滚动规划使得 Rob 局部规划的子目标不断向终点严格趋近.

以下算法的可达性证明部分同静态环境中, 文献[14]对此已进行了详细分析, 在此仅给出相关引理和定理.

**引理 9.** 若存在  $t_w$ , 使  $P_{\text{goal}} \in \text{Win}(P_R(t_w))$ , 则  $\forall t > t_w$ , 都有  $P_{\text{goal}} \in \text{Win}(P_R(t))$ .

**引理 10.** 一定存在有限大小的  $t_w$ , 使  $P_{\text{goal}} \in \text{Win}(P_R(t_w))$ .

**引理 11.** 若  $P_{\text{goal}} \in \text{Win}(P_R(t_w))$ , 则存在有限大小的  $t_g(t_g \geq t_w)$ , 使 Rob 于  $t_g$  到达  $P_{\text{goal}}$ .

**定理 4.** 算法 2 一定能保证 Rob 在有限时间内从起点  $P_{\text{int}}$  安全到达终点  $P_{\text{goal}}$ .

## 5 一类“死锁”问题的解决

在以上分析中,约定动态障碍物不会在  $WS$  中永久静止 ( $\bar{v}_o^i \geq \bar{v}_o$ ). 但在实际环境中,有可能出现动态障碍物突然停止运动,并且处于永久性静止状态的情况,若仍然使用算法 2 就会出现“死锁”现象. 以下将算法 2 加以改进,以推广到存在上述突发事件的环境.

将动态环境约束 A2) 中对  $DObs_i$  速度的约束改为:

$t$  时刻  $DObs_i (i=1, \dots, m)$  的运动速率  $v_o^i(t) \in [0, v_{OH}]$ ; 若要使  $v_o^i(t)=0$ ,  $DObs_i$  需在有限时间内将速率从某一正值减至零; 且一旦  $v_o^i(\tau)=0$ , 则  $\forall t > \tau$ , 都有  $v_o^i(\tau)=0$ .

在新的约束下,  $DObs_i$  可以从运动变为永久静止状态.

令  $\varphi = LFP(P_R(t)P_{sub}(t)) \cap TO_i, i \in D$ , 它表示  $t$  时刻规划的局部可行通道在某动态障碍物预测轨道内的一部分(曲线段).

下面给出算法 3, 其中  $P_{int} \in FD(0), P_{goal} \in SO_i^o, i \in S$ , 并规定  $DObs_i$  不会停留在终点  $P_{goal}$  上.

**算法 3.** 带有事件驱动的滚动路径规划算法.

Step0. 对起点 ( $P_{int}$ )、终点 ( $P_{goal}$ )、工作环境 ( $WS$ )、Rob 的视野半径 ( $r$ ) 进行初始化.

Step1. 如果终点到达, 规划中止.

Step2. 对当前滚动窗口  $Win(P_R(t))$  内的环境信息进行更新.

Step3. 产生局部优化子目标  $P_{sub}(t)$ .

Step4. 在当前滚动窗口内依局部路径行进, 直至以下任一情况发生:

1) 走过  $\epsilon (0 < \epsilon < r)$  长度;

2)  $v_o^i(\tau)=0, \varphi \cap (DO_i(\tau))^o \neq \Phi, i \in DW(\tau), \tau \geq t$ .

Step5. 返回 Step1.

算法中 Step3 产生局部子目标的方法同算法 2 一样, 它只考虑  $Win(P_R(t))$  内探测到的所有静态障碍物, 而不考虑任何动态障碍物的影响. 同样, Step4 中的行进步长  $\epsilon$  不包括因执行局部规划算法 1 的 Step2 中的  $\text{III}$  而引起 Rob 走过的路程.

由算法 3 的 Step4 可知, 该滚动规划蕴含了周期性和事件驱动两种机制. 前者指 Rob 在空间上每行进过固定长度 ( $\epsilon$ ), 便重新进行滚动窗口更新和规划; 后者指局部探测信息发生突变 ( $v_o^i(t)=0$ ), 并危及先前规划路径的安全性时, 不得不重新进行窗口更新和规划 (以后将速率为零的动态障碍物作为静态障碍来处理). 容易证明, 在上述环境约束下引入 Step4 中的 2) 并不影响算法的可达性.

当环境中所有动态障碍物速率都为零时, 算法 3 退化为静态未知环境下的滚动规划, 它只蕴含周期性驱动机制.

## 6 仿真研究

图 4 给出了应用算法 2 对全局环境未知情况下移动机器人路径规划的一个仿真结果. 在规划初始时刻, 机器人不具备任何静、动态障碍物的位置信息. 机器人在未知的动态环境中, 利用实时探测到的局部窗口信息, 进行滚动规划, 在滚动窗口内, Rob 利用算法 1 规划局

部路径. 图中灰色表示未知障碍物, 黑色则表示 Rob 在滚动过程中探测到的局部环境信息, 圆形滚动窗口中心黑点即为 Rob. 图 4(b), (c)和(d)中的滚动窗口内  $DW(t) \neq \Phi$ , 图 4(c)显示了 Rob 运行至动态障碍物轨道边界, 发现间接可逃逸时, 停止运动等待障碍物通过的场景. 可以看出, 采用本文提出的滚动规划算法, Rob 能依较优的规划路径从起点(S)顺利到达终点(G).

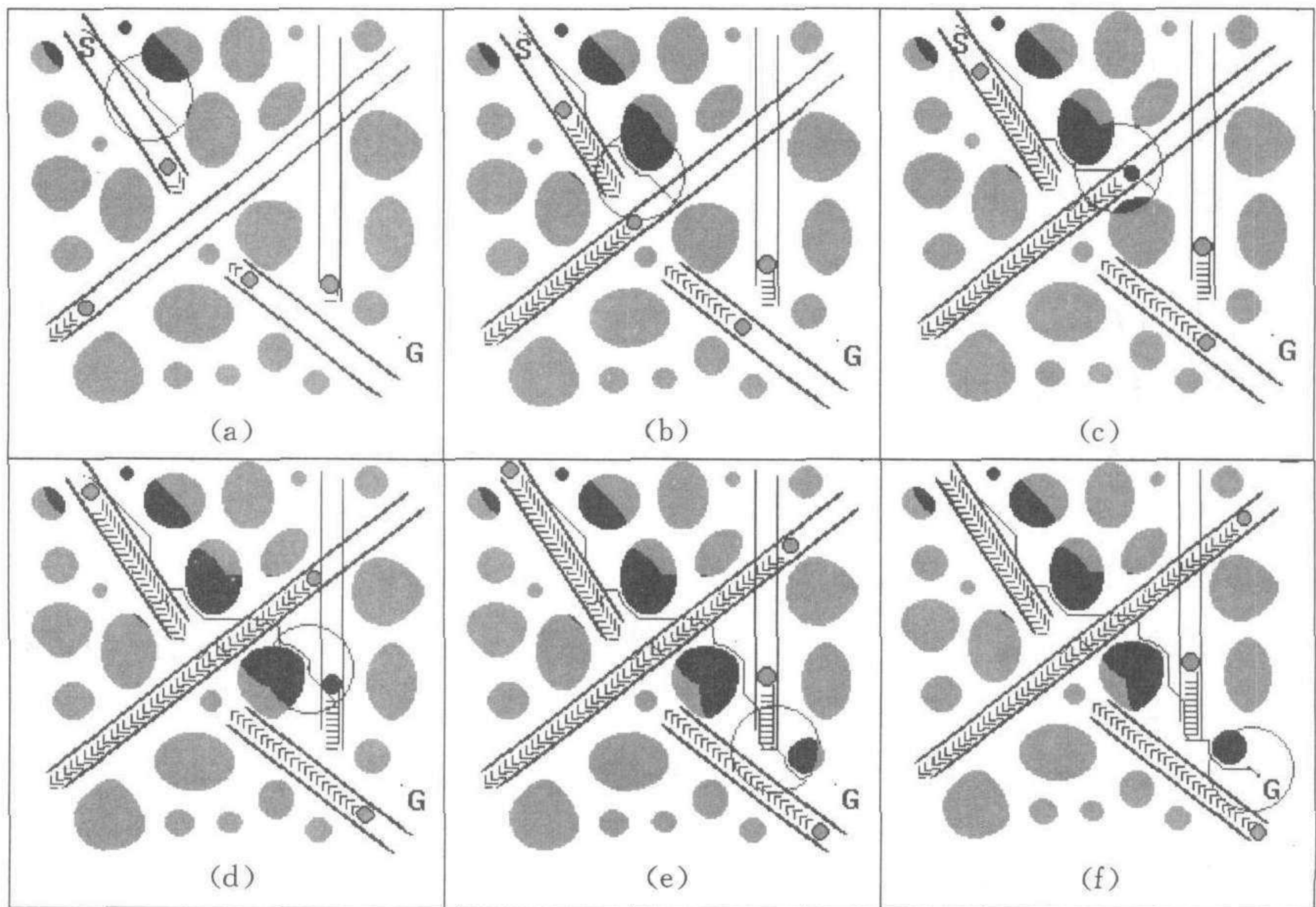


图 4 基于滚动窗的机器人路径规划

作者所进行的大量仿真例子都表明, 基于滚动窗口的路径规划方法具有反应灵敏、实时性好的特点.

## 7 结论

动态未知环境下移动机器人的路径规划是机器人领域的重要研究内容. 由于缺乏先验的全局环境信息, 无法进行有效的离线全局规划. 实际应用中的机器人也往往受到传感范围有限的制约, 没有全局视野, 规划只能依靠实时测得的局部信息. 本文借鉴预测控制的基本原理, 在全局环境信息未知情况下, 用基于滚动窗口的规划方法进行机器人路径规划. 机器人虽然没有先验的全局环境知识, 但充分利用探知的局部信息, 在滚动中有效结合了优化和反馈机制, 不仅使局部规划的计算量保持在较低水平, 而且保证了全局的可达性. 通过本文的研究不难发现, 滚动规划是解决未知环境下规划问题的强有力工具.

## 参 考 文 献

- 1 Schwartz J T, Sharir M. On the "Piano Movers" problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Comm. Pure Appl. Math.*, 1983, **36**(4):345~398
- 2 Lozano-Perez T, Wesley M. An algorithm for planning collision-free paths among polyhedral obstacles. *Comm.*

- ACM, 1979, **22**(6):560~570
- 3 James L C. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, 1985, **1**(1):31~41
  - 4 Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robotics and Automation*, 1991, **7**(3):278~288
  - 5 Lee P S, Wang L L. Collision avoidance by fuzzy logic control for automated guided vehicle navigation. *Journal of Robotic Systems*, 1994, **11**(8):743~760
  - 6 Simmon R. The curvature-velocity method for local obstacle avoidance. In: Proc. IEEE Int. Conf. Robotics and Automation, USA:Minneapolis, MN, 1996. 2275~2282
  - 7 Jing X, Zbigniew M, Lixin Z, et al. Adaptive evolutionary planner/navigator for mobile robots. *IEEE Trans. Evolutionary Computation*, 1997, **1**(1):18~28
  - 8 Abe Y, Shikano M, Fukuda T, et al. Vision based navigation system by variable template matching for autonomous mobile robot. In: Proc. IEEE Int. Conf. Robotics and Automation, Leuven: Belgium, 1998. 952~957
  - 9 Iyengar S S, Jorgensen C C, Rao S V N, et al. Learned navigation paths for a robot in unexplored terrain. In: Proc. 2nd Conf. Artificial Intelligence Applications and Engineering of Knowledge Based Systems, Florida: Miami Beach, 1985. 11~13
  - 10 Lumelsky V J, Mukhopadhyay S, Sun K. Dynamic path planning in sensor-based terrain acquisition. *IEEE Trans. Robotics and Automation*, 1990, **6**(4):462~472
  - 11 Taylor C J, Kriegman D J. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Trans. Robotics and Automation*, 1998, **14**(3):417~426
  - 12 席裕庚. 预测控制. 北京:国防工业出版社, 1993
  - 13 席裕庚. 动态不确定环境下广义控制问题的预测控制. 控制理论与应用, 2000, **17**(5):665~670
  - 14 张纯刚, 席裕庚. 全局环境未知时基于滚动窗口的机器人路径规划. 中国科学(E), 2001, **31**(1):51~58

**席裕庚** 1984年在德国慕尼黑工业大学获得博士学位,现为上海交通大学自动化系教授、博士生导师.主要从事预测控制、复杂系统控制理论和智能机器人的研究.

**张纯刚** 上海交通大学自动化系博士生.研究兴趣为机器人路径规划、多机器人协调规划.

## 中国自动化学会 2002 年部分学术活动

### 中国自动化学会 2002 年重点学术活动计划

项目名称	主要内容	时间	地点	联系人
第八次全国会员代表大会	(1) 审议七届理事会工作报告 (2) 审议七届理事会财务工作报告 (3) 确认章程 (4) 选举八届理事会理事	4 季度	待定	北京中关村南一条一号学会办公室 邮编:100080 电话:62544415 传真:62620908 E-mail: wangh@iamail.ia.ac.cn
八届一次理事会暨学术年会	(1) 选举理事长、副理事长、秘书长、常务理事 (2) 学术活动	11 月	北京	同上
学会工作会议	(1) 学会工作的改革与发展 (2) 学会会员的管理 (3) 学会工作经验介绍	11 月	北京	同上

(下转第 266 页)